RUNNING PYTHON AND JAVA FILES INSIDE A DOCKER CONTAINER

Install docker and run engine then follow the following steps.

Step 1: Create a java/Python file

Create a java or python script that you have to run inside a Docker container.
For example I created 2 files, one FibonacciSeries.java and one test.py python script.

FibonacciSeries.java

```
public class FibonacciSeries {
    public static void main(String[] args) {

        int n = 10;

        int a = 0, b = 1;

        System.out.println("Fibonacci Series up to " + n + " terms:");
        for (int i = 1; i <= n; ++i) {
            System.out.print(a + " ");

            int sum = a + b;
            a = b;
            b = sum;
        }
    }
}
```

test.py

```
print("Hello world")
```

Step 2: Create a Docker file

Next, create a Dockerfile in the same directory as your Python script. The Dockerfile is a text file that contains the instructions for building a Docker image.

For FibonacciSeries.java file:-

```
# Use an official OpenJDK runtime as a parent image
FROM openjdk:latest
# Set the working directory to /app
WORKDIR /app
```

```
# Copy the Java file into the container at /app
COPY FibonacciSeries.java test.py/app
# Compile the Java file
RUN javac FibonacciSeries.java
# Run the compiled Java program when the container launches
CMD ["java", "FibonacciSeries"]
```

For test.py file

```
# Use an official Python runtime as a parent image
FROM python:3.10-slim

# Set the working directory to /app
WORKDIR /app

# Copy the current directory contents into the container at /app
COPY . /app

# Run the command to install any necessary dependencies
RUN pip install --no-cache-dir -r requirements.txt

# Run hello.py when the container launches
CMD ["python", "hello.py"]
```

Step 3: Build a Docker image

Next, to build a docker image we use the following command:

Docker build . //Builds the image without name
Docker build -t <image name> .

For the above files I have done:

Docker build -t myjavaapp . // For building image of java script
Docker build -t mypythonapp // For building image of python script

The `-t` option specifies a tag for the image. The `.` at the end of the command specifies that the build context is the current directory.

Step 4: Run the docker container

To run the docker container we use the following command:
Docker run –rm myjavaapp
Docker run –rm mypythonapp

Other commands:

To list images-
Docker images

```
PS C:\Users\Riya Mary Cleetus\OneDrive\Pictures\Documents\cloudinstitution> docker images
REPOSITORY   TAG       IMAGE ID       CREATED         SIZE
myjavaapp    latest    a206567af26b   23 hours ago    470MB
myapp        latest    719dc556d2ab   23 hours ago    128MB
```

To list containers-
Docker ps -a / Docker ps

```
PS C:\Users\Riya Mary Cleetus\OneDrive\Pictures\Documents\cloudinstitution> docker ps
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS     NAMES
PS C:\Users\Riya Mary Cleetus\OneDrive\Pictures\Documents\cloudinstitution> docker ps -a
CONTAINER ID   IMAGE          COMMAND                  CREATED         STATUS                     PORTS     NAMES
0cefcea51358   myjavaapp      "java FibonacciSeries"   23 hours ago    Exited (0) 23 hours ago              practical_visvesvaraya
b942b7b63353   a206567af26b   "java FibonacciSeries"   23 hours ago    Exited (0) 23 hours ago              wizardly_franklin
ee793a23fc39   a206567af26b   "bash"                   23 hours ago    Exited (0) 23 hours ago              jolly_lalande
0d4bd0d7f7fa   myjavaapp      "java FibonacciSeries"   23 hours ago    Exited (0) 23 hours ago              elastic_noether
```

To login to docker-
Docker login

```
PS C:\Users\Riya Mary Cleetus\OneDrive\Pictures\Documents\cloudinstitution> docker login
Authenticating with existing credentials...
Login Succeeded
```

To delete container-
docker rm <containerid>

```
PS C:\Users\Riya Mary Cleetus\OneDrive\Pictures\Documents\cloudinstitution> docker rm 0cefcea51358
0cefcea51358
```